Experiment 7

Kmeans Clustering of Text corpora

Name:	**:**	**Abhishek N N**	**Register No**	**:**	**20BCE1025**
Faculty	:	Dr. Alok Chauhan	Slot	:	L51-L52 AB1-605B
Course	:	Web Mining Lab	Code	:	CSE3024
Programme	:	B.Tech CSE Core	Semester	:	Win – 22 - 23

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Problem- 1:

Take any text corpora, apply necessary preprocessing and perform the k-means clustering on the corpora.

About the dataset

This contains data of news headlines published over a period of 15 years. From the reputable Australian news source ABC (Australian Broadcasting Corp.) Site: http://www.abc.net.au/ Prepared by Rohit Kulkarni

```
[1] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import seaborn as sns
    from sklearn.feature_extraction import text
    from sklearn.feature_extraction.text import TfidfVectorizer
    from sklearn.cluster import KMeans
    from nltk.tokenize import RegexpTokenizer
    from nltk.stem.snowball import SnowballStemmer
    %matplotlib inline
```

```
[3] data = pd.read_csv("https://raw.githubusercontent.com/franciscadias/data/master/abcnews-date-text.csv",error_bad_lines=False,usecols =["headline_text"])
    data.head()
```

	headline_text
0	aba decides against community broadcasting lic...
1	act fire witnesses must be aware of defamation
2	a g calls for infrastructure protection summit
3	air nz staff in aust strike for pay rise
4	air nz strike to affect australian travellers

```
[4] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1082168 entries, 0 to 1082167
Data columns (total 1 columns):
 #   Column         Non-Null Count    Dtype
---  ------         --------------    -----
 0   headline_text  1082168 non-null  object
dtypes: object(1)
memory usage: 8.3+ MB
```

Deleting dupliate headlines(if any)

```
data[data['headline_text'].duplicated(keep=False)].sort_values('headline_text').head(8)
```

	headline_text
116304	10 killed in pakistan bus crash
57973	10 killed in pakistan bus crash
676588	110 with barry nicholls
673123	110 with barry nicholls
748887	110 with barry nicholls
912413	110 with barry nicholls
898238	110 with barry nicholls episode 15
827356	110 with barry nicholls episode 15

```
[6] data = data.drop_duplicates('headline_text')
```

NLP

```
# preprocessing
# import the necessary libraries
import string
import nltk
import re
def text_lowercase(text):
    return text.lower()
# Remove numbers
def remove_numbers(text):
    result = re.sub(r'\d+', '', text)
    return result
# remove punctuation
def remove_punctuation(text):
    translator = str.maketrans('', '', string.punctuation)
    return text.translate(translator)
# remove whitespace from text
def remove_whitespace(text):
    return  " ".join(text.split())
# remove stopwords
def remove_stopwords(text):
    stopword_list = nltk.corpus.stopwords.words('english')
    tokens = nltk.word_tokenize(text)
    tokens = [token.strip() for token in tokens]
    return ' '.join([token for token in tokens if token not in stopword_list])
# stemming
def stem_text(text):
    ps = nltk.PorterStemmer()
    tokens = nltk.word_tokenize(text)
    tokens = [token.strip() for token in tokens]
    return ' '.join([ps.stem(token) for token in tokens])
# lemmatization
```

```
def lemmatize_text(text):
    wnl = nltk.WordNetLemmatizer()
    tokens = nltk.word_tokenize(text)
    tokens = [token.strip() for token in tokens]
    return ' '.join([wnl.lemmatize(token) for token in tokens])
# remove special characters
def remove_special_characters(text):
    pattern=r'[^a-zA-z0-9\s]'
    text=re.sub(pattern,'',text)
    return text
# remove extra newlines
def remove_extra_newlines(text):
    pattern=r'[\r|\n|\r]+'
    text=re.sub(pattern,' ',text)
    return text
```

```
# apply all the functions to the text
def preprocess(corpus):
    normalized_corpus = []
    # normalize each document in the corpus
    for doc in corpus:
        doc = text_lowercase(doc)
        doc = remove_numbers(doc)
        doc = remove_punctuation(doc)
        doc = remove_whitespace(doc)
        doc = remove_special_characters(doc)
        doc = remove_extra_newlines(doc)
        doc = lemmatize_text(doc)
        doc = stem_text(doc)
        doc = remove_stopwords(doc)
        normalized_corpus.append(doc)
    return normalized_corpus
```

[19]
```
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
True
```

taking 10,000 headlines because of less compute capability

[51]
```
data2=preprocess(data['headline_text'][0:10000])
```

```
data2
```

```
'alic forum discu indigen educ issu',
'alinghi fume anoth cancel',
'least die bu tumbl greec river',
'aussi arriv zimbabw',
'barca deportivo real shine spain',
'bayern open point gap',
'bevan frustrat lack opportun',
'blair put case iraq pope',
'bradman baggi green expect fetch',
'britain ralli support new un resolut',
```

Bag of words

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(data2)
X.toarray()
```

```
array([[0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]])
```

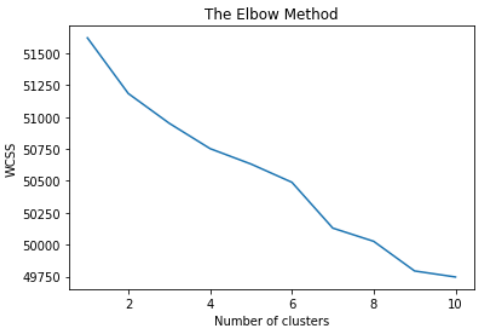
Note: Observe the clustering performance by doing different initializations of centroids as well as different number of clusters.

Elbow method to select number of clusters

This method looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion". This "elbow" cannot always be unambiguously identified. Percentage of variance explained is the ratio of the between-group variance to the total variance, also known as an F-test. A slight variation of this method plots the curvature of the within group variance.

Basically, number of clusters = the x-axis value of the point that is the corner of the "elbow"(the plot looks often looks like an elbow)

```
from sklearn.cluster import KMeans
wcss = []
for i in range(1,11):
    kmeans = KMeans(n_clusters=i,init='k-means++',max_iter=300,n_init=10,random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)
plt.plot(range(1,11),wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.savefig('elbow.png')
plt.show()
```



As more than one elbows have been generated, I will have to select right amount of clusters by trial and error. So, I will showcase the results of different amount of clusters to find out the right amount of clusters.

we see elbow at 6, 7, 8 and 9 it is the best possible range of cluster to form

6 Clusters

```
kmeans = KMeans(n_clusters = 6, n_init = 20) # n_init(number of iterations for clsutering) n_jobs(number of cpu cores to u
kmeans.fit(X)
# We look at 3 the clusters generated by k-means.
common_words = kmeans.cluster_centers_.argsort()[:,-1:-26:-1]
for num, centroid in enumerate(common_words):
    print(str(num) + ' : ' + ', '.join(data2[word] for word in centroid))
```

```
0 : man accus plot shoot parliament, claim u ha secur un vote major medium, call increas fish catch monitor, alston seiz gl
1 : geelong mayor account fraud earn year jail, uni given access qld biggest supercomput, polic investig suspici death new
2 : u send bomber deterr n korea, green gone rest u best seek world berth, hotel bed short suppli hobart, irish presid toas
3 : polic investig suspici death new farm, shire look share health resourc, polic concern acid theft, clijster continu marc
4 : man face committ hear arson charg, thoma thrill ai award, program underway fight park weed, bush assassin saddam report
5 : u poet rise war iraq, storm ravag south east qld, least kill itali car pileup, drought take toll insect, irish presid t
```

7 Clusters

```
[59] kmeans = KMeans(n_clusters = 7, n_init = 20) # n_init(number of iterations for clsutering) n_jobs(number of cpu cores to u
kmeans.fit(X)
# We look at 3 the clusters generated by k-means.
common_words = kmeans.cluster_centers_.argsort()[:,-1:-26:-1]
for num, centroid in enumerate(common_words):
    print(str(num) + ' : ' + ', '.join(data2[word] for word in centroid))
```

```
0 : storm ravag south east qld, drought take toll insect, u poet rise war iraq, least kill itali car pileup, aussi dollar c
1 : geelong mayor account fraud earn year jail, polic investig suspici death new farm, man accus plot shoot parliament, cla
2 : shire look share health resourc, bush assassin saddam report, loss pittman wake call freeman, combin approach tackl gra
3 : stage set khmer roug trial, alston seiz glow telstra report, pentagon list goal war, u poet rise war iraq, protest halt
4 : uni given access qld biggest supercomput, cyclon erica weaken offshor, geelong mayor account fraud earn year jail, bush
5 : u poet rise war iraq, irish presid toast st pat day melbourn, pentagon list goal war, bin laden escap u oper taliban, c
6 : two woman found dead unit, man accus plot shoot parliament, sherpa plan world highest cyber cafe byo oxygen, corium stu
```

8 Clusters

```
kmeans = KMeans(n_clusters = 8, n_init = 20) # n_init(number of iterations for clsutering) n_jobs(number of cpu cores to u
kmeans.fit(X)
# We look at 3 the clusters generated by k-means.
common_words = kmeans.cluster_centers_.argsort()[:,-1:-26:-1]
for num, centroid in enumerate(common_words):
    print(str(num) + ' : ' + ', '.join(data2[word] for word in centroid))
```

```
0 : storm ravag south east qld, drought take toll insect, least kill itali car pileup, u poet rise war iraq, aussi dollar c
1 : claim u ha secur un vote major medium, u plane shot baghdad sabri, man accus plot shoot parliament, alston seiz glow te
2 : u poet rise war iraq, irish presid toast st pat day melbourn, pentagon list goal war, bin laden escap u oper taliban, c
3 : uni lectur investig polit, un border observ withdrawn iraq kuwait, self clone crayfish threaten nativ speci, u send bom
4 : geelong mayor account fraud earn year jail, man accus plot shoot parliament, call increas fish catch monitor, govt oper
5 : polic investig suspici death new farm, polic concern acid theft, atsic chief hinder polic pub brawl court, clijster cor
6 : australia withdraw bougainvil peac monitor, civilian injuri report baghdad, u send bomber deterr n korea, shire look sh
7 : judg hand woman chair fall, india threaten icc suspens, dragila set new indoor pole vault world mark, stage set khmer r
```

9 Clusters

```
[61] kmeans = KMeans(n_clusters = 9, n_init = 20) # n_init(number of iterations for clsutering) n_jobs(number of cpu cores to u
     kmeans.fit(X)
     # We look at 3 the clusters generated by k-means.
     common_words = kmeans.cluster_centers_.argsort()[:,-1:-26:-1]
     for num, centroid in enumerate(common_words):
         print(str(num) + ' : ' + ', '.join(data2[word] for word in centroid))
```

```
0 : shire look share health resourc, runaway bu smash home car, bush assassin saddam report, loss pittman wake call freemar
1 : u poet rise war iraq, man accus plot shoot parliament, claim u ha secur un vote major medium, irish presid toast st pat
2 : call increas fish catch monitor, qld oppn concern race cut, claim u ha secur un vote major medium, u confirm apach heli
3 : storm ravag south east qld, drought take toll insect, u poet rise war iraq, least kill itali car pileup, aussi dollar c
4 : sydney gear mardi gra, council hope water effort flush success, u poet rise war iraq, u send bomber deterr n korea, hop
5 : polic investig suspici death new farm, polic concern acid theft, atsic chief hinder polic pub brawl court, clijster cor
6 : alston seiz glow telstra report, two woman found dead unit, stage set khmer roug trial, sherpa plan world highest cybe1
7 : geelong mayor account fraud earn year jail, uni given access qld biggest supercomput, man whack thatcher get month jail
8 : u stock set open flat, hotel bed short suppli hobart, bin laden escap u oper taliban, u poet rise war iraq, u send bomb
```